Experiment no: 2 Date: 08/08/2024

**Aim:** Write a program in C/Java to implement stack

Write a program in C/Java to reverse a string using stack.

**Software Language**: C

**Theory:** A stack is a linear data structure that follows the principle of **Last In First Out (LIFO)**. This means the last element inserted inside the stack is removed first.

**Basic Operations of Stack:**

There are some basic operations that allow us to perform different actions on a stack.

* Push: Add an element to the top of a stack
  + IsFull: Check if the stack is full(overflow condition)
* Pop: Remove an element from the top of a stack
  + IsEmpty: Check if the stack is empty(underflow condition)
* Peek: Get the value of the top element without removing it

**Algorithm for push operation:**

Step 1 − Checks if the stack is full.(top==max-1) where max is size of array.

Step 2 − If the stack is full, then display “overflow” and exit.

Step 3 − If the stack is not full, increments top to point next empty space.

Step 4 − Adds data element to the stack location, where top is pointing.

Step 5 − Returns success.

**Algorithm for pop operation:**

Step 1 − Checks if the stack is empty. (top == -1)

Step 2 − If the stack is empty, then display “underflow” and exit.

Step 3 − If the stack is not empty, access the data element at which top is pointing.

Step 4 − Decrease the value of top by 1.

Step 5 − Returns success.

**Algorithm for peek operation:**

Step 1 - Check whether stack is EMPTY. (top == -1)

Step 2 - If it is EMPTY, then display "Stack is EMPTY!!!" and terminate the function.

Step 3 - If it is NOT EMPTY, then display top element of the stack.

Step 4 - Return success

**Algorithm to reverse a string:**

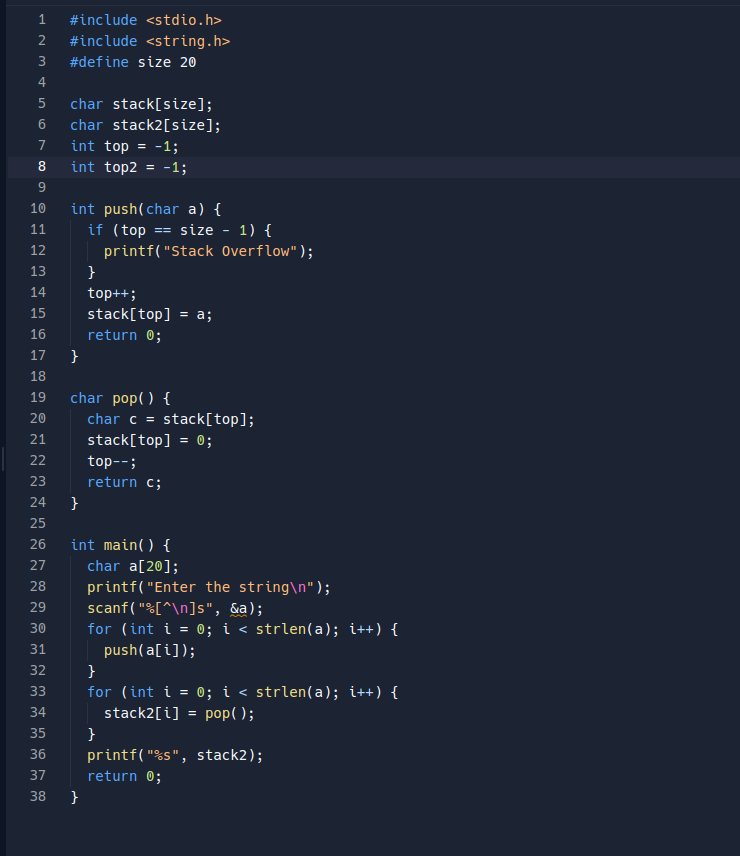
Step 1 – Create an empty stack.

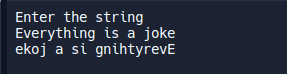
Step 2 - Pick the characters from the string one by one and put them to the stack, so that the last character of the string comes at the top of the stack.

Step 3 - Pop the stack and put the popped characters back in the empty string.

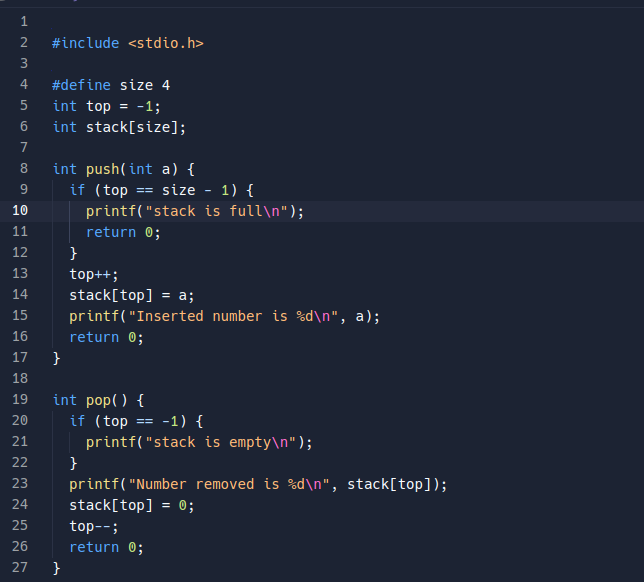
**Code:**

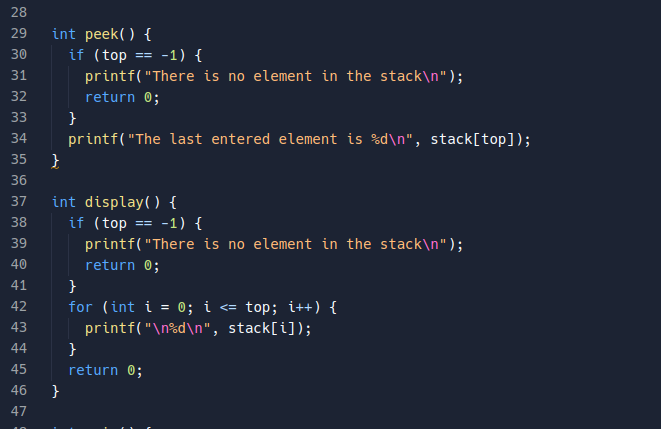
Code and output for reverse a string :

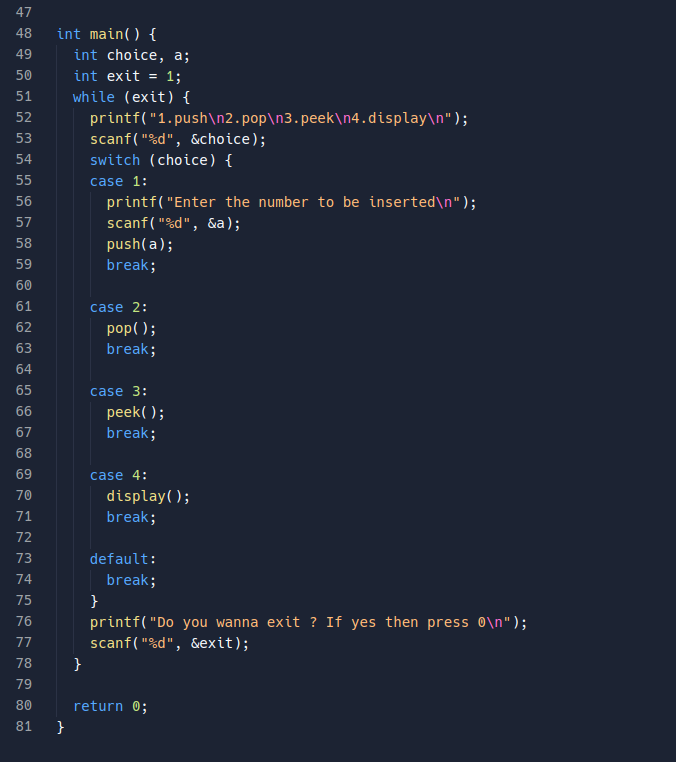


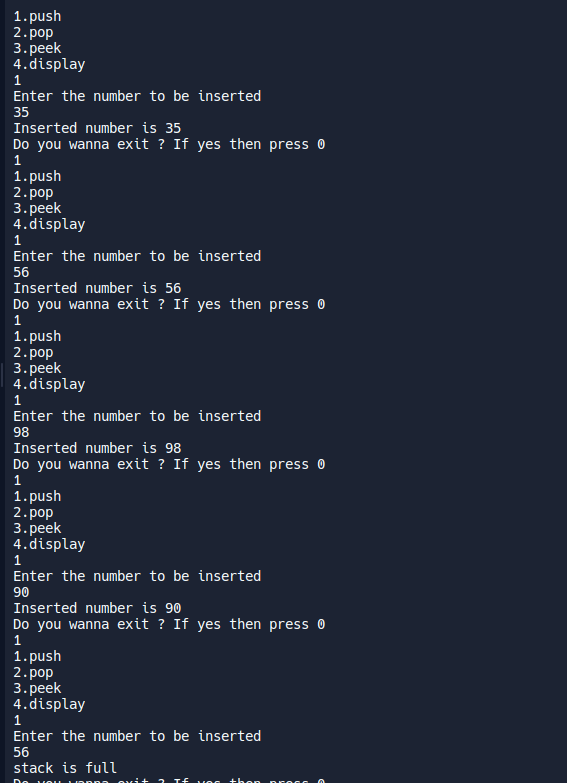


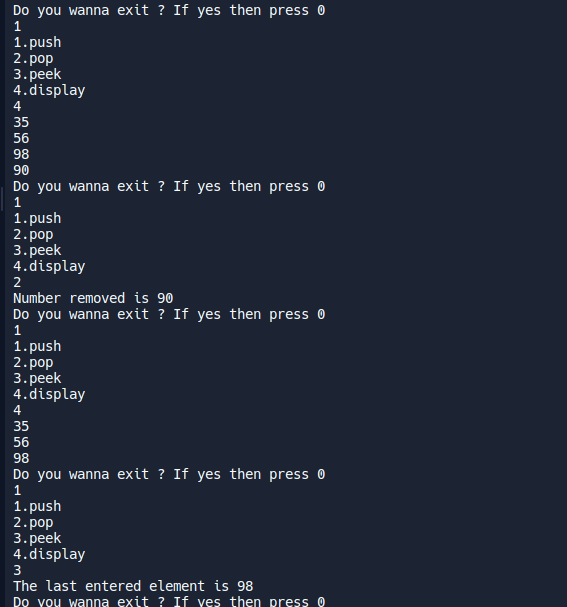
Code and output for push, pop, peek and display algorithm :





****

jay



**Result and Conclusion:**

From the following practical, we learned how the stack works and what are different algorithms or methods used in in stack data structure and how to implement in C.